UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/658,224	09/08/2003	Edward Colton Greene	NVIDP224B/P000872	5350

28875 7590 01/17/2007

Zilka-Kotab, PC
P.O. BOX 721120
SAN JOSE, CA 95172-1120

EXAMINER	
NGUYEN, PHU K	
ART UNIT	PAPER NUMBER
2628	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	01/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/658,224
Filing Date: 09/08/2003
Appellant(s): GREENE ET AL.

Kevin J. Zilka

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed May 10, 2006 appealing from the Office action mailed October 6, 2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(4) Status of Amendments After Final

The statement of the status of claims contained in the brief is not correct since the rejections on claims 20 and 21 are now withdrawn, and these claims are now indicated as allowable.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5999187 DEHMLOW et al. 12-1999

GREENE et al., "Hierarchical Z-Buffer Visibility" Apple Computer, April 1996, pp. 1-7

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-3, 6-9, 12-14, and 16-19, 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over GREENE et al. (Hierarchical Z-buffer Visibility) in view of DEHMLOW et al. (5,999,187).

As per claim 1, Greene teaches the claimed "graphics system including a scene manager, geometric processor means, rendered means, Z-pyramid, and a far clipping plane" (Greene, the culling process of octree cubes of the Z-paramid within the viewing frustum; page 2, column 2 to page 3, column 1, section 3.1, Object space octree). Greene's viewing frustum includes a far clipping plane. It is noted that Greene does not

explicitly teach the far clipping plane is updated "based on the farthest depth value in a depth buffer" as claimed. However, Dehmlow teaches that the updating of far clipping plane of the viewing frustum based on the farthest depth value in said hierarchical depth buffer means is well known in the art (Dehmlow, column 12, lines 43-63; both of near and far clipping planes are dynamically updated based on the closest and farthest cells, respectively). The claimed language does not limit the size of the claimed Z-pyramid, thus for one-level Z-pyramid, Greene's system becomes an octree system as in case of Delmlow (Greene, page 5, column 1, lines 14-16 of section 4.1, "Now that if we shrink the window size down to a single pixel, the hierarchical visibility algorithm becomes a ray caster using an octree subdivision"). It would have been obvious to update the far clipping plane because it reduces the amount of processed data and improves the render speed and the accuracy of the graphics selection mechanism (Dehmlow, column 12, lines 49-53).

As per claim 2, Greene teaches the claimed "graphics system", comprising: a geometric processor; z-pyramid; a renderer; and a far clipping plane" (Greene, the culling process of octree cubes of the Z-paramid within the viewing frustum; page 2, column 2 to page 3, column 1, section 3.1, Object space octree). Greene's viewing frustum includes a far clipping plane. It is noted that Greene does not explicitly teach the far clipping plane is updated "substantially based on a farthest depth value" as claimed. However, Dehmlow teaches that the updating of far clipping plane based on the farthest depth value in said hierarchical depth buffer means is well known in the art (Dehmlow, column 12, lines 43-63; both of near and far clipping planes are dynamically

updated based on the closest and farthest cells, respectively). The claimed language does not limit the size of the claimed Z-pyramid, thus for one-level Z-pyramid, Greene's system becomes an octree system as in case of Delmlow (Greene, page 5, column 1, lines 14-16 of section 4.1, "Now that if we shrink the window size down to a single pixel, the hierarchical visibility algorithm becomes a ray caster using an octree subdivision"). It would have been obvious to update the far clipping plane because it improves the z-buffer resolution and accuracy of graphics selection mechanism (Dehmlow, column 12, lines 49-53).

Claim 3 adds into claim 2 "a scene manager" which Greene teaches in page 5, column 2, Graphics Hardware).

Claim 6 adds into claim 2 "the culling stage is coupled between the geometric processor and the renderer" which Greene teaches in the culling process (page 2, column 2, hierarchical visibility describes the process of culling the hidden objects) in which the culling is performed after the process of inputted geometric data and before the rendering the objects for display.

Claim 7 adds into claim 2 "the far clipping plane is updated based on the farthest depth value" which Greene does not explicitly teach. However, Dehmlow teaches that the updating of far clipping plane based on the farthest depth value in said hierarchical depth buffer means is well known in the art (Dehmlow, column 12, lines 43-63; both of near and far clipping planes are dynamically updated based on the closest and farthest cells, respectively). It would have been obvious to update the far clipping plane

because it improves the z-buffer resolution and accuracy of graphics selection mechanism (Dehmlow, column 12, lines 49-53).

Claims 8-9, 12 claim a method based on the system of claims 1-3, 6-7, therefore, they are rejected under the same reason.

Claims 13-14 claim a computer program product to perform the function of the system of claims 1-3, 6-7 which Greene teaches in the implementation of these function in a software programmed in C (page 71, column 2, Implementation); therefore, they are rejected under the same reason.

Claim 16 adds into claim 2 "the updating includes resetting the far clipping plane to the farthest depth value" " which Greene does not explicitly teach. However, Dehmlow teaches that the updating of far clipping plane based on the farthest depth value in said hierarchical depth buffer means is well known in the art (Dehmlow, column 12, lines 61-63; the farthest point on the fartherst cell is used to position the far clipping plane). The claimed language does not limit the size of the claimed Z-pyramid, thus for one-level Z-pyramid, Greene's system becomes an octree system as in case of Delmlow (Greene, page 5, column 1, lines 14-16 of section 4.1, "Now that if we shrink the window size down to a single pixel, the hierarchical visibility algorithm becomes a ray caster using an octree subdivision"). It would have been obvious to update the far clipping plane because it improves the z-buffer resolution and accuracy of graphics selection mechanism (Dehmlow, column 12, lines 49-53).

Claim 17 adds into claim 2 "the farthest depth value is included in a tip of the z-pyramid" which Green teaches in the page 3, column 2, in which the culling process goes through the coarsest level of the z-pyramid.

Claim 18 adds into claim 17 "the tip of the z-pyramid further includes a coarsest NxN tile in the Z-pyramid" (Green, for an octree spatial subdivision, each level of the Z-paramid comprises NxN tiles in which the tip of the paramid is the coarsest level).

Claim 19 adds into claim 18 "the tip of the z-pyramid further includes additional levels of the z-pyramid" which Green teaches in the page 3, column 2, in which the culling process goes through from the finest to the coarsest level of the z-pyramid.

Claim 22 adds into claim 17 "depth values of the z-pyramid are encoded" which the cited references do not teach. However, the encoded stored data, which is well known in the art, can be applied for data in z-buffer for a motivation of reducing the storage requirement.

Claim 23 adds into claim 22 "the depth values of the z-pyramid are encoded for reducing storage requirements" which would have been obvious because as argued above, the encoded data will be in a compressed form and would need less memory to store them.

Claim 24 adds into claim 2 "the updating accelerates a culling of a box since a depth of a nearest comer of the box is farther than the farthest depth value" which Dehmlow teaches the update (Dehmlow, column 12, lines 43-63; both of near and far clipping planes of the viewing frustum box are dynamically updated based on the closest and farthest cells, respectively) and Green provides a culling of box in Octree,(page 3, column 1). It would have been obvious to update the far clipping plane because it improves the z-buffer resolution and accuracy of graphics selection mechanism (Dehmlow, column 12, lines 49-53).

(10) Response to Argument

Group #1: Claims 1-3, 6-9, 12-14, and 17-19:

Appellant argues "Delmlow relates to a computer aided design (CAD) system while Greene relates to a visibility algorithm. To simply glean features from a CAD system, which aids in the creation and display objects on a computer (such as in Delmlow), and combine the same with the non-analogous art of visibility algorithms (such as in Greene), would simply be improper." Examiner does not agree because Delmlow's CAD system renders the displayed objects using a view frustum culling procedure within a multi level-of-details (LOD) database (Delmlow's octree is an hierarchical data structure of the 3d objects within XYZ space including the depth

dimension; column 2, lines 48-59) within an environment exactly same as Greene's culling/clipping method. Greene, who is also Appellant, taught in his 1996-published technical paper "Hierarchical Z-buffer Visibility" (Greene, page 2, column 2) as following "The **hierarchical Z-buffer visibility algorithm uses an octree spatial subdivision** to exploit object-space coherence, a Z pyramid to exploit image-space coherence, and a list of previously visible octree nodes to exploit temporal coherence. While the full value of the algorithm is achieved by using all three of these together, the object-space octree and the image-space Z pyramid can also be used separately. Whether used separately or together, these data structures make it possible to compute the same result as ordinary Z buffering at less computational expense." Specifically, Greene's visibility algorithm is perfectly compatible for use in Delmlow's system as following: Greene renders the object for its visibility by checking whether the hierarchical octree cubes, which encloses the object or its primitives, intersects the viewing frustum (Greene, page 3, column 1, lines 11-29), whereas Delmlow uses the same viewing frustum 714 for visibility or culling/clipping process (Delmlow, column 11, lines 7-25). Therefore, Delmlow's octree database structure is analogous to Appellant's hierarchical z-buffer.

Appellant further argues that the cited references do not teach "means for updating said far clipping plane based on the farthest depth value in a Z-pyramid, if the farthest depth value in the Z-pyramid is nearer than a depth of the far clipping plane."

In page 3, column 1, Greene taught that "From this observation, the basic algorithm is easy to construct. We begin by placing the geometry into an octree,

associating each primitive with the smallest enclosing octree cube. Then we start at the root node of the octree and render it using the following recursive steps: First, we check to see if the octree cube intersects **the viewing frustum**. If not, we are done. If the cube does intersect the viewing frustum, we scan convert the faces of the cube to determine whether or not the whole cube is hidden. If the cube is hidden, we are done. Otherwise, we scan convert any geometry associated with the cube and then recursively render its children in front-to-back order." In the virtual 3D space where the observer sees the 3D image corresponds to a 3D image display region; this display region is represented as **a viewing frustum consisting of 6 planes including the farthest and nearest depth planes** in which the observer sees the 3D image within the viewing frustum through the screen of the CRT and only an object within the viewing frustum appears on the screen. Delmlow's **viewing frustum** 714 is a box bounded within six planes including the far clipping plane (i.e., plane 715f). Delmlow specifically emphasizes that "the near and far clipping planes are positioned dynamically based on the part(s) (i.e., the cells that have non-null cell-to-art mapping) that are present within the five-point view frustum boundary 714 (column 12, lines 43-46)"; which means the updates of the close and far clipping planes 715e and 715f are based on the cells that have non-null cell-to-art mapping. More specifically, Delmlow states "the farthest point on the farthest cell (again with a non-null cell-to-part mapping) is used to position the far clipping plane" (column 12, lines 61-63); which means "whenever the farthest depth value does not match (i.e., nearer or farther) the current position of the far clipping plane, the far clipping plane will be updated to the farthest

depth value;" or in a Z-pyramid as claimed "updating said far clipping plane based on the farthest depth value, if the farthest depth value is nearer than a depth of the far clipping plane."

Moreover, in a special case of one-level Z_pyramid (the claimed language does not limit the size of the claimed Z-pyramid), Greene's hierarchical z-buffer system becomes an octree system as in case of Delmlow (Greene, page 5, column 1, lines 14-16 of section 4.1, "Now that if we shrink the window size down to a single pixel, the hierarchical visibility algorithm becomes a ray caster using an octree subdivision"). In this special case, not only Greene's hierarchical z-buffer system is analogous to Delmlow's hierarchical octree system, but also their dynamically updated viewing frustum is "updating said far clipping plane based on the farthest depth value, if the farthest depth value is nearer than a depth of the far clipping plane."

In conclusion, either in a one-level or multi_level general Z-pyramid, it would have been obvious, in view of Delmlow's dynamic updated viewing frustum, to cinfigure Greene's viewing frustrum as claimed by having its far clipping plan "updated based on the farthest depth value, if the farthest depth value is nearer than a depth of the far clipping plane." The motivation of updating the farthest and nearest depth plans on the viewing frustum is to reduce the amount of data outside the viewing range to improve the rendering speed.

Group #2: Claim 16

Appellant argues, "Examiner has not taken the forgoing claim language of claim 16 into context. Specifically, the claimed 'farthest depth value' is specifically included in a Z-pyramid as set forth in intervening claim 1."

In page 3, column 1, Greene taught that "From this observation, the basic algorithm is easy to construct. We begin by placing the geometry into an octree, associating each primitive with the smallest enclosing octree cube. Then we start at the root node of the octree and render it using the following recursive steps: First, we check to see if the octree cube intersects **the viewing frustum**. If not, we are done. If the cube does intersect the viewing frustum, we scan convert the faces of the cube to determine whether or not the whole cube is hidden. If the cube is hidden, we are done. Otherwise, we scan convert any geometry associated with the cube and then recursively render its children in front-to-back order." In the virtual 3D space where the observer sees the 3D image corresponds to a 3D image display region; this display region is represented as **a viewing frustum consisting of 6 planes including the farthest and nearest depth planes** in which the observer sees the 3D image within the viewing frustum through the screen of the CRT and only an object within the viewing frustum appears on the screen. Delmlow's **viewing frustum** 714 is a box bounded within six planes including the far clipping plane (i.e., plane 715f). Delmlow specifically emphasizes that "the near and far clipping planes are positioned dynamically based on the part(s) (i.e., the cells that have non-null cell-to-art mapping)

that are present within the five-point view frustum boundary 714 (column 12, lines 43-46)"; which means the updates of the close and far clipping planes 715e and 715f are based on the cells that have non-null cell-to-art mapping. More specifically, Delmlow states "the farthest point on the farthest cell (again with a non-null cell-to-part mapping) is used to position the far clipping plane" (column 12, lines 61-63); which means "whenever the farthest depth value does not match (i.e., nearer or farther) the current position of the far clipping plane, the far clipping plane will be updated to the farthest depth value;" or in a Z-pyramid as claimed "updating said far clipping plane based on the farthest depth value, if the farthest depth value is nearer than a depth of the far clipping plane."

Moreover, in a special case of one-level Z_pyramid (the claimed language does not limit the size of the claimed Z-pyramid), Greene's hierarchical z-buffer system becomes an octree system as in case of Delmlow (Greene, page 5, column 1, lines 14-16 of section 4.1, "Now that if we shrink the window size down to a single pixel, the hierarchical visibility algorithm becomes a ray caster using an octree subdivision"). In this special case, not only Greene's hierarchical z-buffer system is analogous to Delmlow's hierarchical octree system, but also their dynamically updated viewing frustum is "updating said far clipping plane based on the farthest depth value, if the farthest depth value is nearer than a depth of the far clipping plane."

In conclusion, either in a one-level or multi_level general Z-pyramid, it would have been obvious, in view of Delmlow's dynamic updated viewing frustum, to cinfigure Greene's viewing frustrum as claimed by having its far clipping plan "updated based on

the farthest depth value, if the farthest depth value is nearer than a depth of the far clipping plane." The motivation of updating the farthest and nearest depth plans on the viewing frustum is to reduce the amount of data outside the viewing range to improve the rendering speed.

Group #3: Claim 20

Examiner agrees with Appellant's arguments, and would like to indicate allowable subject matter in claim 20.

Group #4: Claim 21

Examiner agrees with Appellant's arguments, and would like to indicate allowable subject matter in claim 21.

Group #5: Claims 22-23

Appellant argues, "wherein depth values of the Z-pyramid are encoded" (claim 22) and "wherein the depth values of the Z-pyramid are encoded for reducing storage requirements thereof" (see claim 23). Delmlow teaches the encoding of depth values to save the storage requirements for Z-values (Delmlow, 3D run length encoded format or different formats; column 7, lines 11-12, or 51-56).

Group #6: Claim 24

Appellant argues, "wherein the updating accelerates a culling of a box since a depth of the nearest corner of the box is farther the farthest depth value." Delmlow's octree box is used in the comparison of the viewing frustum (column 11, lines 15-19) in which the culling process accelerates in case of the octree box is outside the viewing frustum when a depth of the nearest corner of the box is farther the farthest depth value.

For conclusion, the rejection of claims 1-3, 6-9, 12-14, 16-19, 22-24 are maintained, whereas claims 20 and 21 are allowable if rewritten in independent forms.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Phu K. Nguyen

PHU K. NGUYEN
PRIMARY EXAMINER
GROUP 2300

Conferees:
Michael Razavi MR
R. Eisenzopf
Jeffery A. Brier JB